E.G.S.PILLAY ENGINEERING COLLEGE

(Autonomous)

Approved byAICTE,New Delhi|Affiliated to AnnaUniversity, Chennai AccreditedbyNAAC with „A"Grade|Accredited byNBA (CSE, EEE, MECH, ECE, CIVIL, IT)

NAGAPATTINAM–611002

B.E. – Computer Science and Engineering

2019 Regulation:Full Time Curriculum and Syllabus

SEMESTER V									
Course Code	**Course Name**	**L**	**T**	**P**	**C**	**Maximum Marks**			**Category**
						CA	**ES**	**Total**	
Theory Course									
1901MA501	Discrete Mathematics	3	2	0	4	40	60	100	BS
1902CS501	Object Oriented Analysis and Design	3	0	0	3	40	60	100	PC
1902CS502	Theory of Computation	3	2	0	4	40	60	100	PC
	PC Elective I - Computer Graphics and Multimedia	3	0	0	3	40	60	100	PE
	HSS Elective I- Total Quality Management	3	0	0	3	40	60	100	HSSE
1901MCX03	Essence of Indian Traditional Knowledge	2	0	0	0	100	0	100	MC
Laboratory Course									
1902CS551	CASE Tools Laboratory	0	0	2	1	50	50	100	PC
1902CS552	Mobile Application Development Laboratory	0	0	2	1	50	50	100	PC
1904GE551	Life Skills: Aptitude I	0	0	2	1	100	-	100	EEC
Total		**17**	**4**	**6**	**20**	**500**	**400**	**900**	

L–Lecture|T–Tutorial|P–Practical|C–Credit|CA –ContinuousAssessment| ES–EndSemester

1901MA501	DISCRETE MATHEMATICS	L	T	P	C
		3	0	0	3

PREREQUISITES:
Engineering mathematics I and II

COURSE OBJECTIVES:

1. Get familiar and understand the fundamental notation in discrete mathematics.
2. Explore the concepts of counting principle and graph theory.
3. Understand and demonstrate the basic concept of an algorithm and its application in combinatorial mathematics.

Module I	SET THEORY AND LOGIC	9 Hours

Sets, function, relation, equivalence relation, Poset, Function logic, Proposition logic, Predicates and quantifiers - Nested quantifiers – Rules of inference - Proofs methods and strategy.

Module II	INDUCTION AND COMBINATORICS	9 Hours

Mathematical induction – In the basics of counting – The pigeon hole principle – Permutation and Combination - Solving linear recurrence relation – Generating function – Principle of inclusion an exclusion.

Module III	GRAPH	9 Hours

Graph – Sub graphs – Operation on graph – Matrix representation of graph, path and connectedness – Graph isomorphism – Euler and Hamilton's paths and graph.

Module IV	ALGEBRAIC STRUCTURE	9 Hours

Algebraic system – Semi groups, Monoids, Groups , Subgroups and their properties – Cyclic groups – Cosets – Permutation groups - Lagrange's theorem – Cayley's theorem – Normal subgroups, Homomorphism of groups – Introduction to rings and fields.

Module V	LATTICES AND BOOLEAN ALGEBRA	9 Hours

Lattices aspartially order sets, properties of lattices – lattices as algebraic system some special lattices – Boolean algebra.

TOTAL:	45 HOURS

COURSE OUTCOMES:

	On completion of the course the students will be able to
CO1	Distinguish between the notion of discrete and continuous mathematical structures.
CO2	Solve the linear recurrence relation using generating function.
CO3	Construct finite state diagram using graph representation.
CO4	Apply mathematical foundations, algorithms, principle and computer science theory in the modeling and design of computer based system.
CO5	Understand the concept and significance of lattice and Boolean algebra in computer science.

REFERENCES

1. Kenneth H, Rosen – Discrete mathematics and its applications, 7th edition, Tata MC grow hill, Delhi 2012.
2.C.L.liu and D.P.Mohapatra – Elements of discrete mathematics, A computer oriented approach, MC grow Hill, Third edition, 2012.
3.Ralph.p, Grimaldi – Discrete and combinatorial mathematics, An applied introduction – Fourth edition person education Asia, Delhi 2002.
4.Trembly J.P and Manohar R – Discrete mathematical structure with application to computer science, Tata MC grow hill, Delhi.
5.Peter J Cameron – combinatorics – Topics, Technique and algorithms, Cambridge University Press.
6.Nptel.ac.in/course/111105035 www.nptel videos in 2012/11/mathematics.

1902CS501	OBJECT ORIENTED ANALYSIS AND DESIGN	L	T	P	C
		3	0	0	3

PREREQUISITES:
1. Software Engineering
2. Programming Concepts

COURSE OBJECTIVES:

1. To develop background knowledge as well as core expertise in object oriented System.
2. To provide the importance of the software design process.
3. Learn the basics of OO analysis and design skills the UML design diagrams.

Module I	UML DIAGRAMS	9 Hours

Introduction to OOAD – Unified Process - UML diagrams – Use Case – Class Diagrams– Interaction Diagrams – State Diagrams – Activity Diagrams – Package-Component and Deployment Diagrams

Module II	DESIGN PATTERNS	9 Hours

Object oriented design methodology – GRASP: Designing objects with responsibilities – Patterns– Creator – Information expert – Low coupling –Controller – High cohesion – Designing for visibility - GoF design patterns – Adapter – Singleton – Factory – Strategy – Composite - Facade and observer patterns

ModuleIII	APPLYING DESIGN PATTERNS	9 Hours

System sequence diagrams - Relationship between sequence diagrams and use cases Logical architecture - UML package diagram – UML class diagrams - UML interaction diagrams - Finding conceptual classes and description classes – Associations – Attributes – Domain model refinement – Finding conceptual class Hierarchies - Aggregation and Composition.

Module IV	IMPLEMENTATION AND APPLICATION	9 Hours

Forward Engineering – Reverse Engineering - Test driven development – Refactoring – UML tools and UML as blueprint - UML state machine diagrams and modeling - UML deployment and component diagrams

Module V	CODING AND TESTING	9 Hours

Mapping design to code – Testing: Issues in OO Testing – Class Testing – OO Integration Testing – GUI Testing – OO System Testing.

TOTAL:	45 HOURS

FURTHER READING / CONTENT BEYOND SYLLABUS / SEMINAR:

1. Advanced Design Patterns.
2. Developing SRS Documents.
3. Case Studies: Various tools used in OOAD.

COURSE OUTCOMES:

	After completion of the course, Student will be able to
CO1	Create use case documents that capture requirements for a software system.
CO2	Use the UML analysis and design diagrams.
CO3	Apply design patterns that facilitate development and evolution of new models.
CO4	Address the real-world problems by modeling software solutions using UML tools.
CO5	Compare and contrast various testing techniques.

REFERENCES:

1. Craig Larman, "Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development", Third Edition, Pearson Education, 2015.
2. Micheal Blaha, James Rambaugh, "Object-Oriented Modeling and Design with UML", Second Edition, Prentice Hall of India Private Limited, 2007
3. Mike O'Docherty, "Object-Oriented Analysis & Design: Understanding System Development with UML 2.0", John Wiley & Sons, 2005.
4. James W- Cooper, Addison-Wesley, "Java Design Patterns – A Tutorial", 2000.
5. Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, "Design patterns: Elements of Reusable object-oriented software", Addison-Wesley, 2009
6. http://nptel.ac.in/courses/106105153/

1902CS502	THEORY OF COMPUTATION	L	T	P	C
		3	2	0	4

PREREQUISITES:

1. Basic Programming Knowledge
2. Basic Mathematical Knowledge

COURSE OBJECTIVES:

1. Introduce students to the mathematical foundations of computation including automata theory; the theory of formal languages and grammar; the notions of algorithm, decidability, complexity and computability
2. Enhance / develop student's ability to understand and conduct mathematical proofs for computation and algorithms.
3. Be able to construct Turing machines and Post machines.
4. Understand the notions of decidability and undecidability of problems, Halting problem.

Module I	**FINITE AUTOMATA**	**9+3 Hours**

Introduction- Basic Mathematical Notation and techniques- Finite State systems – Basic Definitions – Finite Automaton – DFA & NDFA – Finite Automaton with Є- moves – Regular Languages- Regular Expression – Equivalence of NFA and DFA – Equivalence of NDFA"s with and without Є-moves – Equivalence of finite Automaton and regular expressions –Minimization of DFA– Pumping Lemma for Regular sets – Problems based on Pumping Lemma.

Module II	**GRAMMARS**	**9+3 Hours**

Grammar Introduction– Types of Grammar – Context Free Grammars and Languages– Derivations and Languages – Ambiguity- Relationship between derivation and derivation trees – Simplification of CFG – Elimination of Useless symbols – Module productions – Null productions – Greiback Normal form – Chomsky normal form – Problems related to CNF and GNF.

Module III	**PUSHDOWN AUTOMATA**	**9+3 Hours**

Pushdown Automata- Definitions – Moves – Instantaneous descriptions – Deterministic pushdown automata – Equivalence of Pushdown automata and CFL – pumping lemma for CFL – problems based on pumping Lemma

Module IV	**TURING MACHINES**	**9+3 Hours**

Definitions of Turing machines – Models – Computable languages and functions –Techniques for Turing machine construction – Multi head and Multi tape Turing Machines – The Halting problem – Partial Solvability – Problems about Turing machine- Chomskian hierarchy of languages..

Module V	**UNSOLVABLE PROBLEMS AND COMPUTABLE FUNCTIONS**	**9 +3Hours**

Unsolvable Problems and Computable Functions – Primitive recursive functions – Recursive and recursively enumerable languages – Universal Turing machine. MEASURING AND CLASSIFYING COMPLEXITY: Tractable and Intractable problems- Tractable and possibly intractable problems – P and NP completeness – Polynomial time reductions.

	Total:	**45 + 15 Hours**

FURTHER READING / CONTENT BEYOND SYLLABUS / SEMINAR :

1. Introduction to Infinite Automata Theory
2. Advanced theory of computation.

COURSE OUTCOMES:

At the end of the course, the student should be able to:

CO1	Demonstrate advanced knowledge of formal computation and its relationship to languages.
CO2	Distinguish different computing languages and classify their respective types.
CO3	Recognize and comprehend formal reasoning about languages.
CO4	Show a competent understanding of the basic concepts of complexity theory.
CO5	To understand the concept of turing machine.

TextBooks:

1. Hopcroft J.E., Motwani R. and Ullman J.D, "Introduction to Automata Theory, Languages and Computations", Second Edition, Pearson Education, 2008. (MODULE 1,2,3)
2. John C Martin, "Introduction to Languages and the Theory of Computation", Third Edition, Tata McGraw Hill Publishing Company, New Delhi, 2007. (MODULE 4,5)

REFERENCES
1. Mishra K L P and Chandrasekaran N, "Theory of Computer Science – Automata, Languages and Computation", Third Edition, Prentice Hall of India, 2004.
2. Harry R Lewis and Christos H Papadimitriou, "Elements of the Theory of Computation", Second Edition, Prentice Hall of India, Pearson Education, New Delhi, 2003.
3. Peter Linz, "An Introduction to Formal Language and Automata", Third Edition, Narosa Publishers, New Delhi, 2002.
4.Kamala Krithivasan and Rama. R, "Introduction to Formal Languages, Automata Theory and Computation", Pearson Education 2009
5.http://nptel.ac.in/downloads/106106049/

1902MCX03	ESSENCE OF INDIAN TRADITIONAL KNOWLEDGE	L	T	P	C
		2	0	0	0

MODULE I	**INTRODUCTION TO CULTURE**	**6 Hours**
Culture, civilization, culture and heritage, general characteristics of culture, importance of culture in human literature, Indian Culture, Ancient India, Medieval India, Modern India		
MODULE II	**INDIAN LANGUAGES, CULTURE AND LITERATURE**	**6 Hours**
Indian Languages and Literature-I: the role of Sanskrit, significance of scriptures to current society, Indian philosophies, other Sanskrit literature, literature of south India Indian Languages and Literature-II: Northern Indian languages & literature		
MODULE III	**RELIGION AND PHILOSOPHY**	**6 Hours**
Religion and Philosophy in ancient India, Religion and Philosophy in Medieval India, Religious Reform Movements in Modern India (selected movements only)		
MODULE IV	**FINE ARTS IN INDIA (ART, TECHNOLOGY& ENGINEERING)**	**6 Hours**
Indian Painting, Indian handicrafts, Music, divisions of Indian classic music, modern Indian music, Dance and Drama, Indian Architecture (ancient, medieval and modern), Science and Technology in India, development of science in ancient, medieval and modern India		
MODULE V	**EDUCATION SYSTEM IN INDIA**	**6 Hours**
Education in ancient, medieval and modern India, aims of education, subjects, languages, Science and Scientists of Ancient India, Science and Scientists of Medieval India, Scientists of Modern India.		
	TOTAL	**30 Hours**

REFERENCES:

1. KapilKapoor, “Text and Interpretation: The India Tradition”,ISBN: 81246033375, 2005
2. “Science in Samskrit”, SamskritaBharti Publisher, ISBN 13: 978-8187276333, 2007
3. NCERT, “Position paper on Arts, Music, Dance and Theatre”, ISBN 81-7450 494-X, 200
4. S. Narain, “Examinations in ancient India”, Arya Book Depot, 1993
5. SatyaPrakash, “Founders of Sciences in Ancient India”, Vijay Kumar Publisher, 1989
6. M. Hiriyanna, “Essentials of Indian Philosophy”, MotilalBanarsidass Publishers, ISBN 13: 978- 8120810990, 2014

1902CS551	CASE TOOLS LAB	L	T	P	C
		0	0	2	1

PREREQUISITES:
1. Software Engineering
2. Programming Concepts

COURSE OBJECTIVES:

1. To highlight the importance of object-oriented analysis and design and its limitations.
2. To show how we apply the process of object-oriented analysis and design to software development.
3. To provide the necessary knowledge and skills in using object-oriented CASE tools.

LIST OF EXPERIMENTS:

1. To develop a problem statement and statement of work.
2. Develop an IEEE standard SRS document. Also develop risk management and project plan
3. Identify Use Cases and develop the Use Case model.
4. Identify the business activities and develop an UML Activity diagram.
5. Identity the conceptual classes and develop a domain model with UML Class diagram.
6. Using the identified scenarios find the interaction between objects and represent them using UML Interaction diagrams
7. Draw the State Chart diagram.
8. Identify the User Interface, Domain objects, and Technical services. Draw the partial layered, logical architecture diagram with UML package diagram notation
9. Draw Component and Deployment diagrams.
10. Practice forward engineering and reverse engineering

TOTAL: 45 HOURS

SUGGESTED DOMAINS FOR MINI PROJECT

1. Passport Automation System
2.Automatic Teller Machine
3. Book bank System
4.Exam Registration
5. Stock maintenance system.
6.E-ticketing
7.Software Personnel management System
8.Recruitment System

ADDITIONAL EXPERIMENTS / INNOVATIVE EXPERIMENTS:

1. Credit card Processing
2. Library Management System.

SUGGESTED SOFTWARE TOOLS: Rational Suite (or) Argo UML (or) equivalent, Eclipse IDE and JModule

COURSE OUTCOMES:

After completion of the course, Student will be able to

CO1	Design and implement projects using OO concepts.
CO2	Recognize the role and function of each UML model in developing object-oriented software.
CO3	Apply appropriate design patterns.
CO4	Create code from design.
CO5	Compare and contrast various testing techniques

REFERENCES:

1. Manual Prepared by the course instructor
2. Craig Larman, "Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development", Third Edition, Pearson Education, 2015.
3.http://www.seminarsonly.com/computer%20science/itwin-seminar-report-ppt-pdf.php

1902CS552	MOBILE APPLICATION DEVELOPMENT LAB	L	T	P	C
		0	0	2	1

PREREQUISITES: Computer Programming Languages: Java

COURSE OBJECTIVES:

1. To explore about the structure of mobile development framework
2. To analyze the issues of mobile application
3. To develop the dynamic application using various parts of android projects

List of Experiments:

1. Develop an interactive application with different layout managers
2. Develop Applications with Multiple Activities and a Simple Menu using various View options
3. Develop an application for calculator operation
4. Develop an application that implements multi thread concepts
5. Develop an application using all Google map API functionalities
6. Develop an dynamic application that implements database manipulation
7. Develop an media oriented application using A/V function
8. Develop an application that writes data to the SD card.
9. Develop an application that creates an alert upon receiving a message.
10. Develop an sensor based application for ballgame sensor

Total: 30 Hours

Additional Experiments:

1. Develop an application that makes use of RSS Feed.
2. Write a mobile application that creates alarm clock.

COURSE OUTCOMES:

After completion of the course, Student will be able to

CO1	To understand the working of mobile application development
CO2	To paraphrase the multiple activity options in one application
CO3	To understand the background data processing about the application
CO4	To analyze the inter-thread communication between the activities and functions
CO5	To describe about the sensor implementation in android

REFERENCES:

1. Android 6 for Programmers: An App-Driven Approach by Paul J. Deitel , Harvey Deitel , Alexander WaldPrentice Hall; 3 edition 2015
2. Android Application Development in 24 Hours, by Carmen Delessio , Lauren Darcey , Shane ConderSams Publishing; 4 edition 2015
3. Android Cookbook: Problems and Solutions for Android Developers by Ian Darwin Shroff/O'Reilly; Second edition 2017
4. Beginning Android Programming with Android Studio by J. F. DiMarzio Wiley publication Fourth edition 2016

1904GE551	LIFE SKILLS: APTITUDE – 1	L	T	P	C
		0	0	2	1

COURSE OBJECTIVES:

- To brush up problem solving skill and to improve intellectual skill of the students
- To be able to critically evaluate various real life situations by resorting to Analysis Of key issues and factors
- To be able to demonstrate various principles involved in solving mathematical problems and thereby reducing the time taken for performing job functions.
- To enhance analytical ability of students
- To augment logical and critical thinking of Student

MODULE I	INTRODUCTION TO NUMBER SYSTEM, BASIC SHORTCUTS OF ADDITION, MULTIPLICATION, DIVISION	6 Hours

Classification of numbers – Types of Numbers - Divisibility rules - Finding the units digit - Finding remainders in divisions involving higher powers - LCM and HCF Models - Fractions and Digits – Square, Square roots – Cube, Cube roots – Shortcuts of addition, multiplication, Division.

MODULE II	RATIO AND PROPORTION, AVERAGES	6 Hours

Definition of Ratio - Properties of Ratios - Comparison of Ratios - Problems on Ratios - Compound Ratio - Problems on Proportion, Mean proportional and Continued Proportion Definition of Average - Rules of Average - Problems on Average - Problems on Weighted Average - Finding average using assumed mean method.

MODULE III	PERCENTAGES, PROFIT AND LOSS	6 Hours

Introduction Percentage - Converting a percentage into decimals - Converting a Decimal into a percentage - Percentage equivalent of fractions - Problems on percentages - Problems on Profit and Loss percentage- Relation between Cost Price and Selling price - Discount and Marked Price - Two different articles sold at same Cost Price - Two different articles sold at same Selling Price - Gain% / Loss% on Selling Price.

MODULE IV	CODING AND DECODING, DIRECTION SENSE	6 Hours

Coding using same set of letters - Coding using different set of letters - Coding into a number - Problems on R-model - Solving problems by drawing the paths - Finding the net distance travelled - Finding the direction - Problems on clocks - Problems on shadows - Problems on direction sense using symbols and notations.

MODULE V	NUMBER AND LETTER SERIES NUMBER AND LETTER ANALOGIES, ODD MAN OUT	6 Hours

Difference series - Product series - Squares series - Cubes series - Alternate series - Combination series - Miscellaneous series - Place values of letters - Definition of Analogy - Problems on number analogy - Problems on letter analogy - Problems on verbal analogy - Problems on number Odd man out - Problems on letter Odd man out - Problems on verbal Odd man out

TOTAL	30 Hours

COURSE OUTCOMES:

On the successful completion of the course, students will be able to

CO1	Learners should be able to understand number and solving problems least time using various shortcuts
CO2	Compare two quantities using ratio and proportion, Solve problems on Partnership, Mixture & Allegation and ages least time using shortcuts and apply real life situations
CO3	Learns should be able to understand the concept behind Average and Percentage.
CO4	Workout concepts of Coding and Decoding, ability to visualize directions and understand the logic behind a sequence.
CO5	Learners should be able to find a series the logic behind a sequence.

REFERENCES:

1.Arun Sharma, 'How to Prepare for Quantitative Aptitude for the CAT', 7th edition, McGraw Hills publication, 2016.

2.Arun Sharma, 'How to Prepare for Logical Reasoning for CAT', 4th edition, McGraw Hills publication, 2017.

3.R S Agarwal, 'A modern approach to Logical reasoning', revised edition, S.Chand publication, 2017.

4.R S Agarwal, 'Quantitative Aptitude for Competitive Examinations', revised edition, S.Chand publication,2017.

5.Rajesh Verma, "Fast Track Objective Arithmetic", 3rd edition, Arihant publication,2018.

6.B.S. Sijwalii and InduSijwali, “A New Approach to REASONING Verbal & Non-Verbal”, 2nd edition, Arihnat publication, 2014.

ASSESSMENT PATTERN :

1. Two tests will be conducted (25 * 2) - 50 marks
2. Five assignments will be conducted (5*10) - 50 Marks.